Fast light field inpainting propagation using angular warping and color-guided disparity interpolation

Pierre Allain, Laurent Guillo, Christine Guillemot

Fast light field inpainting propagation using angular warping and color-guided disparity interpolation

Pierre Allain[1], Laurent Guillo[2], Christine Guillemot[1⋆]

[1] Inria Rennes Bretagne-Atlantique, 263 Avenue du Général Leclerc, 35042 Rennes
[2] University of Rennes 1, CNRS, Irisa, 35000 Rennes, France

Abstract. This paper describes a method for fast and efficient inpainting of light fields. We first revisit disparity estimation based on smoothed structure tensors and analyze typical artefacts with their impact for the inpainting problem. We then propose an approach which is computationally fast while giving more coherent disparity in the masked region. This disparity is then used for propagating, by angular warping, the inpainted texture of one view to the entire light field. Performed experiments show the ability of our approach to yield appealing results while running considerably faster.

1 Introduction

As the capture of 4D light fields from real scenes is gaining in popularity, the need for efficient editing tools is expected to rise as well. However, the very large volume of data which they represent, as well as the need to maintain texture and structure consistency across views, raise challenging complexity issues in these processing tasks.

In this paper, we focus on light field inpainting for applications such as object removal. Although the problem of image inpainting has already been extensively studied, only a few methods in the literature address the specific case of 4D light fields. One can find some work on the related topic of multiview image inpainting, however the methods as in [1], [2] are designed for captures with large baselines and do not generalize well for dense light fields.

It is only recently that methods have been proposed for dense light fields inpainting. The authors in [3] extend 2D patch-based methods to 4D patches and ensure consistency by minimizing a 4D patch bi-directional similarity measure. This method progresses patch per patch in a greedy fashion and suffers from a high computational complexity. In addition, the global consistency of the entire light field is not guaranteed. In [4], the central view is first edited using a 2D patch-based method, and the offsets between the filled patches and their best

⋆ This work was supported by the EU H2020 Research and Innovation Programme under grant agreement No 694122 (ERC advanced grant CLIM)

match in the known region are propagated to the other views. But the above methods suffer from high complexity and/or from angular view coherency.

Inpainting first one view (in general the center one) and then coherently propagating the inpainted texture to the other views is an interesting approach as it allows using any 2D inpainting method on the center view to have a good initialization. This is related to the general problem of edit propagation [5]. In this category of methods, the approach in [6] inpaints the center view using a 2D patch-based method and the other views are inpainted by searching for the best matching patches in the inpainted center view. However, because of the greedy nature of the algorithm and the fact that the views are processed separately, the inpainting results may contain angular inconsistencies. The authors in [7] exploit smoothness priors in the epipolar plane images (EPI) to coherently propagate information across views using PDE-based diffusion, while a method based on matrix completion is proposed in [8], assuming a low rank prior for the light field data.

One critical issue for propagating texture to all views is the estimation of disparity in the masked region. In [9], the disparity map is inpainted using a linear interpolation followed by variational regularization. However, an interpolation per EPI may lead to spatial inconsistencies in the disparity maps. In [7], dominant orientations of known pixels are computed as weighted averages of structure tensors across all the views of a given slice. Dominant orientations of unknown pixels are found using a super-pixel guided interpolation.

In this paper, we focus on the fast propagation of the inpainted texture across the entire light field. For sake of computational efficiency, we consider a simple warping strategy which requires knowledge of scene geometry in the inpainted region, that is to say the disparity at this location, as shown in Fig. 1. We revisit the problem of disparity estimation using smoothed structure tensors. We show that, while smoothing generally increases spatio-temporal coherence of estimated disparity fields, it may also lead to misalignment between color and disparity, and in turn to incoherent disparity in the masked region. A direct structure tensor computation is used instead and it is shown that, even if the disparity field is much noisier, this noise does not affect the inpainted disparity values as it turns out to be automatically removed via the color-guided interpolation. This simple approach reduces computational complexity while improving disparity coherency in the masked region. The inpainted disparity allows us to perform a simple angular warping to propagate inpainted texture across the whole light field.

In summary, the contributions of the paper are as follows: 1) A fast disparity estimation of a masked area guided by an inpainted view. 2) The ability of applying inpainting on any view of the light field with corresponding disparity at angular position, since disparity is computed exactly on the view (no averaging). 3) A direct propagation of inpainted area over the whole light field using a 2D forward warping method. 4) An overall speed increase as compared to other methods.

The paper is organized as follows: Section 3 and 4 describe the selected disparity computation method and the method for interpolating the disparity inside the inpainted area. Section 5 presents the method used for propagating a view across the light field, and in Section 6 we discuss some experimental results.

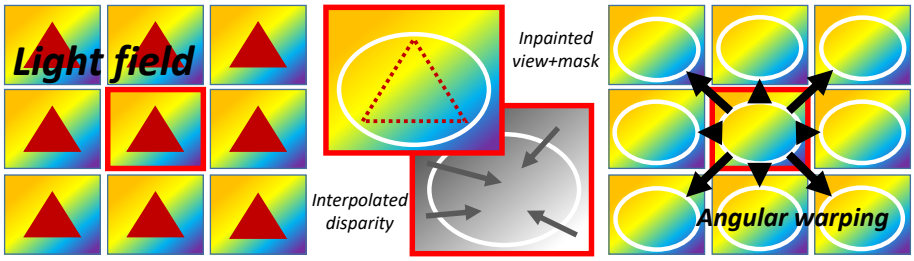


Fig. 1: Overview of the method. The red triangle represents inpainted object in the scene. Corresponding mask standing for inpainted area is delimited by the white line. Scene geometry (depth) behind the mask is unknown. Disparity is therefore interpolated according to subapertures color features.

2 Disparity estimation

Let $L(x, y, u, v)$ denote the 4D representation of a light field, describing the radiance of a light ray parameterized by its intersection with two parallel planes [10], and where (u, v) denote the angular (view) coordinates and (x, y) the spatial (pixel) coordinates. While disparity estimation methods based on structure tensors [11] usually operate on EPI smoothed with a 2D Gaussian kernel, we consider instead a direct computation of the structure tensors on the 4D light field L as

$$\mathbf{T}_{xu} = \nabla_{xu} L \otimes \nabla_{xu} L^{\intercal} * G_{\sigma}(x, y, u, v), \tag{1}$$

where $\nabla_{xu} L = \begin{pmatrix} \partial_x L \\ \partial_u L \end{pmatrix}$, denotes the (x, u) gradient of L, and where $G_{\sigma}(x, y, u, v)$ is a Gaussian kernel smoothing along the four light field dimensions.

The orthogonal eigenvectors θ_+ and θ_- with respective eigenvalues λ_+ and λ_- (where $\lambda_+ > \lambda_-$) of $\mathbf{T}_{xu}$ give a robust computation of the dominant gradient orientations. We are interested in the eigenvector θ_- with the smallest eigenvalue which corresponds to the isophote lines whose slopes give the amount of disparity between the views. Same applies for computing the vertical disparity with the tensor $\mathbf{T}_{yv}$.

While the use of a smoothing kernel yields disparities with enhanced spatial coherence, this also leads to mis-alignment (some cross overs) between the disparity estimates and objects contours (Fig. 2). We will see in the next section that such cross overs can produce bad estimates of disparity inside the inpainted area.

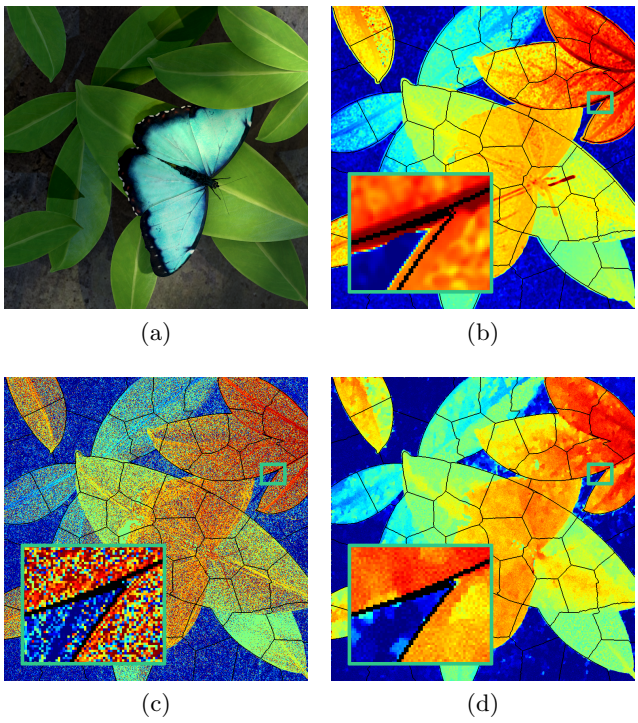


Fig. 2: Disparity estimate of one light field view (a), with black lines accounting for original image's segmentation of dominant shapes. (b) Using Sobel gradient and smoothing tensor with a Gaussian kernel of size 7. (c) Without gradient or tensor smoothing. (d) Without smoothing + TVL1. Despite being nosier, estimate without smoothing better fits object contours.

The structure tensor in the approach is therefore computed with no smoothing, and in order to remove most of outliers estimates, i.e. reducing the noise, we apply a Total Variation (TV-L1) denoising [12] on the disparity estimate:

$$D = \frac{1}{2}\left(\frac{\theta_{xu-}\overrightarrow{x}}{\theta_{xu-}\overrightarrow{u}} + \frac{\theta_{yv-}\overrightarrow{y}}{\theta_{yv-}\overrightarrow{v}}\right), \tag{2}$$

searching to optimize:

$$\tilde{D} = arg\,min_{\tilde{D}}||\nabla\tilde{D}||_1 + \lambda||\tilde{D} - D||_2, \tag{3}$$

where λ is the total variation regularization parameter and was set to 0.5 in our experiments. This value yields a good trade-off regarding smoothness of the regularization.

We finally obtain a disparity estimate specific to inpainted view, being spatially accurate regarding scene contours, and in a very fast way. In the next section, we will address the problem of both filling inpainted mask and dealing with remaining noise.

3 Superpixel guided disparity interpolation

In order to propagate the inpainted area across all views, we need a disparity information inside the mask that does not exist in the light field scene, since inpainting a view can be seen as modifying depth geometry in the inpainted area. As proposed in [7], we use a superpixel segmentation of the inpainted view to guide disparity inpainting, making the assumption that disparity (i.e. depth) is homogeneous in local regions having the same color. The computed superpixels $S = \{S_s\}$, with $\{s \in \mathbb{N} | s \leq Q\}$, are first merged such as any of them lying even partly inside the masked area has at least K pixels outside the mask, i.e. known data. The number of superpixels is defined such as $Q = \frac{1}{\gamma}P$, P being the number of pixels in each subaperture. In our experiments γ was set to 1000 pixels per superpixel, which allows to capture main structures of the image, and K to 20, so that a certain area of each superpixel exists outside the mask.

The disparity $\hat{D}$ for any pixel at position x inside the mask and belonging to a superpixel S_s is estimated by interpolating known disparity values $\tilde{D}$ of $W \leq K$ closest pixels outside the mask but from the same superpixel as:

$$\hat{D}(x) = \frac{\Sigma_i^W w_{s,i}(x, y_i)\tilde{D}(y_i)}{\Sigma_i^W w_{s,i}(x, y_i)}, \tag{4}$$

where $y_i \in S_s$ is a pixel position outside the mask. We also set W to 20 in our experiments. The weights are defined as:

$$w_s(x, y_i) = exp\left(-\frac{(d_s(x, y_i) - \mu_s(x))^2}{\sigma_s(x)^2}\right), \tag{5}$$

$$\text{with:} \quad d_s(x, y_i) = \|x - y_i\|^2 + (I(x) - I(y_i))^2 f_s \tag{6}$$

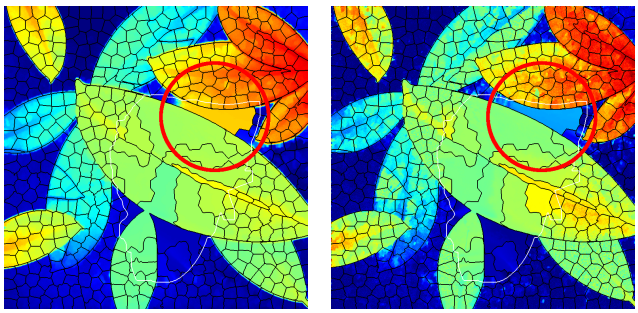
accounting for the spatial distance between the two pixels (inside and outside the mask) and of the color proximity of the two pixels in the inpainted view.

Knowing that the spatial distance $d_s(x, y_i)$ can greatly vary for the different pixels x within a given superpixel, we apply in Eq.(5) a centering of weights around the minimum distance as:

$$\begin{cases} \mu_s(x) = \min(\{d_{s,i}\}), \\ \sigma_s(x)^2 = \beta \text{Var}(\{d_{s,i}\}). \end{cases} \tag{7}$$

Hence, pixels with known disparity and closest to the border of the mask have a higher weight. The scaling between spatial and color distance in Eq.(6 is defined by a function specific to superpixel geometry as $f_s = \alpha \frac{\text{Var}(\{x\})}{\text{Var}(\{I(x)\})}$, $x \in S_s$ to compensate for higher spatial and color variance inside some superpixels. In our experiments, we set $\alpha = 10^{-5}$ and $\beta = 1$ which yielded a good trade-off for interpolation smoothness and coherence with outer mask disparity.

Superpixel guided interpolation has proved to be effective for disparity reconstruction [7]. It is nonetheless very sensitive to disparity cross overs. In Fig. 3a, we can see that the disparity interpolation uses in some cases irrelevant



(a) (b)

Fig. 3: Disparity estimate of inpainted view with focus on cross over issue, white line accounting for inpainting mask boundaries. a) As in [7] for a vertical EPI., b) Our method. Highlighted superpixel mostly use information close to the leaf due to mask configuration. Disparity retrieval is therefore extremely sensitive to estimate around object edges.

information because of mask and segmentation configuration. In Fig. 3b, this reconstruction error is strongly alleviated because of the disparity cross over avoidance introduced in Section 2.

As for the rest of the inpainted area, interpolation weights guided by superpixels prove to be enough for obtaining an homogeneous disparity reconstruction despite noisy inputs, in the same fashion as in [13]. Finally, we obtain a consistent disparity estimate inside the mask which is derived from the inpainted view only and not by using information from the other views, with a few computations and approximations.

4 Inpainting propagation

The computed disparity in the masked area is used to propagate the inpainted area of one view to all the other views of the light field using an angular forward warping. Instead of propagating information on one angular dimension in an EPI framework, we chose to warp directly in directions (u, v). Two main advantages of such approach are as follows. First, we do not need to repeat the whole pipeline of disparity computation and inpaint propagation for each EPI stripe $N+1$ times (for a light field of size $N \times N$ views). And second, direct warping is very fast as compared to diffusion.

However, warping has well known drawbacks. Backward warping implies to know disparity in each destination view, i.e. computing $N^2 - 1$ disparity estimates, which is time consuming. Forward warping on the other hand only needs disparity at the origin of propagation, but comes with ghosting and cracking effects. We propose a solution to tackle the latter issues in the scope of fast light field inpainting.

Handling ghosting artefacts: Ghosting effects, which are classical in depth image-based rendering, arise when multiple pixels with different depth and color converge to a same one in the warped image, resulting in a mean value. Ghosting

artifacts are often avoided by detecting depth discontinuities, in order to separate the boundary layer (containing pixels near the contours of an object) from the main layer (containing pixels far from the contours) [14]. The main layer is first projected then the boundary layer is added everywhere it is visible (i.e. where its depth value is smaller than the main layers one). Ghosting artifacts can also be avoided by estimating background and foreground contributions in the rendered view with the help of advanced matting techniques [15], or using a confidence measure for each projected pixel as in [16].

For sake of reduced complexity, we consider instead a simple disparity mean filter to classify the projected pixels within the mask area as locally belonging to the foreground or the background. This simple mean depth classification approach gives satisfactory results in the case where the number of depth layers does not exceed two. However, in the case of more complex scenes with a higher number of layers, approaches such as z-buffering would be needed to avoid ghosting artefacts. Nonetheless, one can note that for dense light fields with small disparities, as those considered here, such case remains marginal.

Let us suppose that two pixels coming from two different objects with different depths are projected on the same pixel, when doing the texture propagation. Instead of handling a z-buffer to order the projected pixels from foreground to background, thus saving computationally expensive sorting operations, we compute the mean disparity of pixels projected on the same point. We then classify pixels having a disparity higher than the mean as foreground occluding pixels, and those having a disparity smaller than the mean as occluded pixels. In order to obtain such mean disparity, we simply warp the disparity by itself, see Fig. 4. The color is then warped with pixels classified as occluding pixels overlaying those classified as occluded pixels. This simple method also prevents aliasing of warping by still averaging occluding pixels that converge to the same point. This method is directly applied in the two angular dimensions (u, v) of light field, which also provides better continuity of propagation as compared to a sequential one dimension u, v projection.

Handling cracks: Cracking effects, oppositely, arises where a destination pixel does not receive any information due to a local divergence of disparity. This issue is dealt with the help of unsupervised inpainting as in [17] and yields satisfying results for tested light fields.

Eventually, the computational cost of inpainting propagation for one view accounts for two forward warpings and an inpainting of cracks. The later is negligible provided cracks are small, which was the case in our tests.

5 Computational complexity analysis

Algorithmic complexity is presented for the differents stages of the method. First, disparity estimate obtained by Eqs.(1, 2) is computed all over the view at which the inpainting is assigned. The estimate is regularized using Eq.(3). Both algorithms have $O(P)$ complexity.

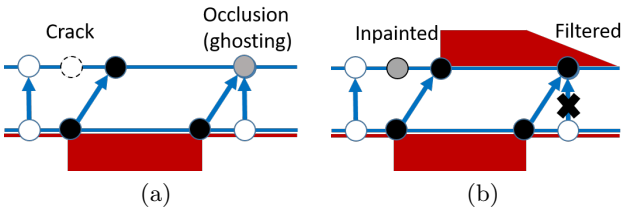


Fig. 4: Forward warping in 1D. Circles stand for image pixels. Disparity is represented in red. a) Base algorithm, b) Modified algorithm with warping of disparity by itself.

Disparity interpolation guided by inpainted view: SLIC algorithm [18] used for superpixel segmentation has $O(P)$ complexity. Fast implementations of the algorithm exist, and make in practice superpixel segmentation computational cost to be low. Merging superpixels with respect to the mask has an approximate complexity $O(R\log(R))$, with R being the number of superpixels not containing at least K pixels outside the mask. This process therefore strongly depends on image segmentation and mask characteristics, as well as on value K. Assuming R is mostly dependant on the number of superpixels Q, we obtain $O(Q\log(Q))$ complexity for the merging step. In our experiments, for a total number of superpixels around 400 (depending on the dataset), computation took in most complex cases only a few seconds. Disparity interpolation inside the mask defined at Eq.(4) has complexity $O(MW)$, with M being the number of pixels in the mask. It is, in most cases, more costly than the last two processes.

Inpainting propagation: As presented in section 4, two forward warpings of inpainted view are performed in our method. Each of them as $O(N^2M)$ complexity. The second one also involve unsupervised inpainting of cracks. This last step has $O(V)$ complexity, whith V being the number of *cracked* pixels. Their number is usally low for dense light fields, therefore the cost of this step is neglectible.

Overall: The global method then has $O(P)$, $O(Q\log(Q))$, $O(MW)$ and $O(N^2M)$ complexities. In our experiments, we found that increasing Q and W does not noticeabily change inpainting results. Therefore, our method's complexity can be approximated by $O(N^2M)$. However, in case a small inpainting mask is applied, disparity estimate and superpixel segmentation become dominant, and the complexity is then $O(P)$.

6 Experimental results

Fig. 5 shows an inpainting result in comparison with method [6]. The experiments were performed on real light fields captured by camera Lytro 1.0. To alleviate color fluctuations between views, we first perform a histogram matching between the central and remaining views. The algorithm takes as input the mask of the area to be inpainted in the whole light field and an inpainted view (usually the central one), which here has been inpainted using the patch-based method of [19]. We compare our method to [7], for which we perform 100 iterations of diffusion in the reference algorithm. Computations for the later and ours were performed

using a C++ implementation on a i7-6600U without any parallelization and the execution times are presented in Tab. 1.

Our method performs the fastest inpainting for almost every dataset tested. Because the mask applied on light field *Bee2* has small size, method of [8] is faster. Indeed, the singular value decomposition used has a $O(N^4M)$ complexity linearily dependent on mask's size, whereas our method has $O(P)$ complexity in this case.

Let us note that from a memory allocation perspective, our method consumes only a few calculation images. For comparison [8] requires approximately $8M(N^2 + 400)$ floating point numbers to store in memory.

More results with videos showing the inpainting consistency across views are provided at `https://www.irisa.fr/temics/demos/lightField/InpaintFast/index.html`, and Fig. 6 shows additional results comparison to state of the art methods.

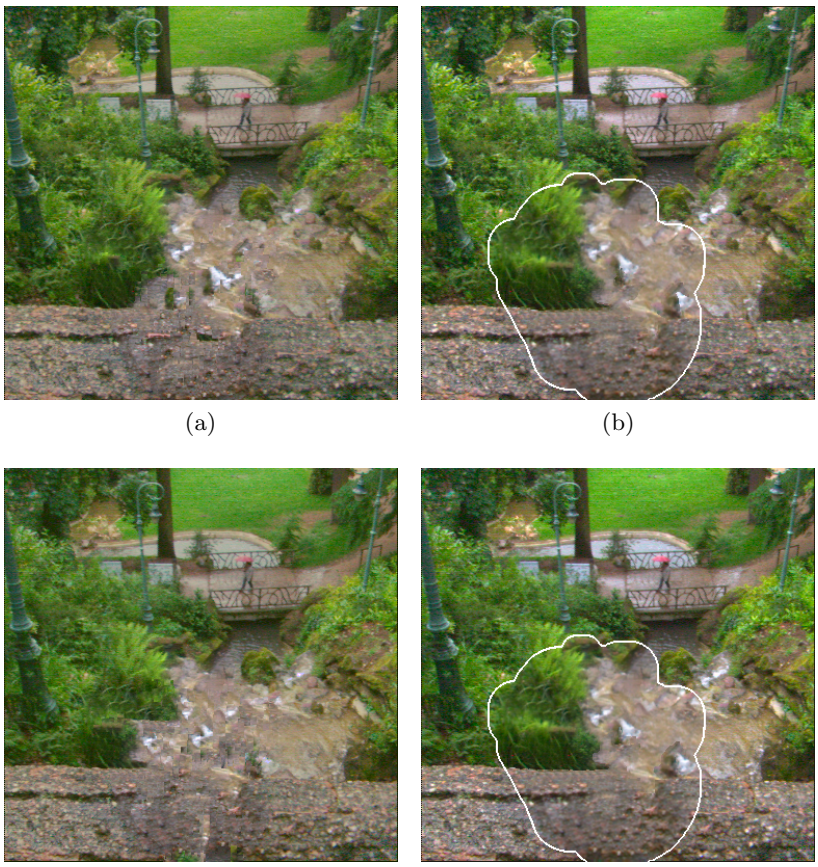
Experiments show very similar results regarding refocusing on focus plane of removed object and its corresponding background, while running considerably faster than other methods.

Dataset	N	Resolution	Mask	[7]	[8]	Our method
Bee1	9	625*434	4.4%	1'29"	-	4"6
Bee2	11	625*434	1.6%	1'17"	4"2	5"3
Figurines	11	625*434	17.4%	3'39"	19"3	7"6
Fruits	9	625*434	4.8%	1"38	-	4"6
Totoro Park	7	379*379	17.2%	50"	9"2	2"5
Totoro Waterfall	7	379*379	24%	1'2"	9"1	3"
Tape	7	379*379	17.1%	1"12	-	2"6
Still life	9	768*768	4.6%	4'2"	13.2"	11"9
Butterfly	9	768*768	21.6%	6'17"	45"	19"3

Table 1: Execution times. Our new method performs with an average of ~ 21 times faster than proposed in [7], and ~ 2.2 times than [8]. Regarding the later, our method performs faster for larger inpainting masks. As compared to [6], our experiments on Lytro Illum light fields ran around ~ 1.3 times faster, without GPU implementation.

7 Conclusion

In this paper we have presented a new fast approach for inpainting light fields. Our method is based on a minimum of relevant computation steps. By first obtaining a coarse disparity estimate at inpainted view coordinates in angular space, we are able to interpolate the disparity at inpainted object's location. This allows a fast propagation of inpainted view using specifically adapted warping techniques. Results quality match state of the art, while running most of the time significantly faster. As a future work, our algorithm would highly benefit of parallel programming like GPU implementation.



(a) (b) (c) (d)

Fig. 5: Inpainting of *Totoro waterfall* light field. a,b) Top left view. c,d) Bottom right view. a,c) With method proposed by [6]. See the incoherence of the texture in the mask area. b-d) Using our method. Views are more coherent along angular coordinates inside the inpainting area with the proposed algorithm.

References

1. S.-H. Baek, I. Choi, and M. H. Kim, "Multiview image completion with space structure propagation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
2. T. Thonat, E. Shechtman, S. Paris, and G. Drettakis, "Multi-view inpainting for image-based scene editing and rendering," in *Fourth International Conference on 3D Vision (3DV)*, Oct 2016, pp. 351–359.
3. K.-W. Chen, M.-H. Chang, and Y.-Y. Chuang, "Light field image editing by 4D patch synthesis," in *IEEE International Conference on Multimedia and Expo, ICME*, June 2015, pp. 1–6.

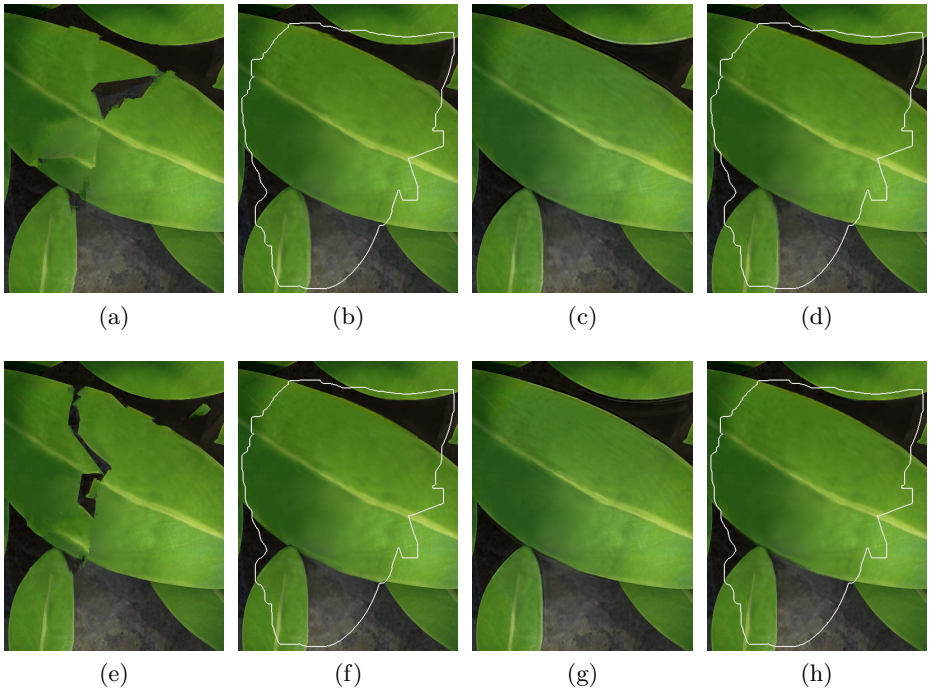


Fig. 6: Inpainting of *Butterfly* light field. a-d) Top left view. e-h) Bottom right view. a,e) With method proposed by [6]. See the incoherence of the texture in the mask area. b,f) With method proposed by [7]. See misalignment of the edges of the central leave. c,g) With method proposed by [8] as reference result. d,h) Using our method which result matches reference.

4. F. L. Zhang, J. Wang, E. Shechtman, Z. Y. Zhou, J. X. Shi, and S. M. Hu, "Plenopatch: Patch-based plenoptic image manipulation," *IEEE Trans. on Visualization and Computer Graphics*, vol. 23, no. 5, pp. 1561–1573, Feb. 2016.
5. A. Jarabo, B. Masia, and D. Gutierrez, "Efficient propagation of light field edits," in *Proc. of the V Ibero-American Symposium in Computer Graphics*, ser. SIACG 2011, 2011, pp. 75–80.
6. Williem, K. W. Shon, and I. K. Park, "Spatio-angular consistent editing framework for 4D light field images," in *Multimedia Tools and Applications*, vol. 745, 2016, pp. 16 615–16 631.
7. O. Frigo and C. Guillemot, "Epipolar plane diffusion: An efficient approach for light field editing," London, France, Sep. 2017.
8. M. L. Pendu, X. Jiang, and C. Guillemot, "Light field inpainting propagation via low rank matrix completion," *IEEE Transactions on Image Processing*, vol. PP, no. 99, pp. 1–1, 2018.
9. B. Goldluecke and S. Wanner, "The variational structure of disparity and regularization of 4D light fields," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 1003–1010.
10. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph." *23rd Annual Conference on Computer Graphics and Interactive Techniques, ACM*, pp.

43–54, 1996.
11. S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, 2013.
12. L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259 – 268, 1992.
13. D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast global image smoothing based on weighted least squares," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5638–5653, Dec 2014.
14. C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, pp. 600–608, August 2004.
15. S. Hasinoff, S. Kang, and R. Szeliski, "Boundary matting for view synthesis," *Comput Vis Image Understanding*, vol. 103, p. 2232, 2006.
16. V. Jantet, C. Guillemot, and L. Morin, "Joint Projection Filling method for occlusion handling in Depth-Image-Based Rendering," *3D Research*, pp. http://vincent.jantet.free.fr/publication/jantet–11–3DResearch.pdf, 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00628019
17. A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004.
18. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.
19. M. Daisy, D. Tschumperlé, and O. Lézoray, "A fast spatial patch blending algorithm for artefact reduction in pattern-based image inpainting," in *SIGGRAPH Asia 2013 Technical Briefs*, ser. SA '13. New York, NY, USA: ACM, 2013, pp. 8:1–8:4. [Online]. Available: http://doi.acm.org/10.1145/2542355.2542365